

TP2: DISEÑO DE UN LENGUAJE DE PROGRAMACIÓN

Grupo 10: Nicolás R. Kalaydjian, Eitan Wajsberg, Nataly Sofia Rohling Insua, Nehuen Balian Amaros,
Rocío Ochoa Charlín, Guido Marchi y Julian Alejandro Mosquera

CONTENIDOS

- 01/ Categorías léxicas.
- 02/ Categorías sintácticas.
- 03/ Programa de ejemplo.
- 04/ Derivación de prueba.
- 05/ Conclusiones.

01/ CATEGORÍAS LÉXICAS

Operadores = \+ | * | =

Identificador = Mayúscula (Mayúscula + GuiónBajo)*

GuiónBajo = _

Mayúscula = A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z

Minúscula = a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z

ConstanteEntera = 0 | NroNoCero

NroNoCero = Signo DígitoNoCero (DígitoNoCero | 0)*

Signo = ε | -

DígitoNoCero = 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

Cadena = “(Minúscula + \s)*”

CaracteresPuntuación = \(|\|)

PalabrasReservadas = ent | cad | mostrar | ing

02/

CATEGORÍAS SINTÁCTICAS

programa: listaDeclaraciones(op) listaSentencias(op)

*listaDeclaraciones: sentDeclaración
sentDeclaración listaDeclaraciones*

sentDeclaración: tipoDato Identificador valorAsignado(op)

valorAsignado: = valorAAsignar

valorAAsignar: (uno de) ConstanteEntera Cadena exprAritmética

tipoDato: (uno de) ent cad

(Si tipoDato es **ent** entonces valorAAsignar debe ser ConstanteEntera o exprAritmética. En caso contrario, será una Cadena)

02/

CATEGORÍAS SINTÁCTICAS

exprAritmética: término

exprAritmética + término

término: factor

*termino * factor*

factor: ConstanteEntera

Identificador

(exprAritmética)

(El identificador debe ser previamente declarado como tipo **ent**)

02/ CATEGORÍAS SINTÁCTICAS

listaSentencias: sentIO

sentIO listaSentencias

sentIO: (uno de) sentIngreso sentSalida

sentIngreso: ing Identificador

sentSalida: mostrar salida

salida: (uno de) Identificador ConstanteEntera Cadena exprAritmetica

(El identificador debe ser previamente declarado)

03/ PROGRAMA DE EJEMPLO

cad HOLA = "hola"

ent NUMUNO = 12

ent NUMDOS = 5

ent NUMTRES = NUMUNO + NUMDOS

ent NUMCUATRO

mostrar "ingresar nro entero"

ing NUMCUATRO

mostrar NUMTRES + NUMCUATRO

04/ DERIVACIÓN DE PRUEBA

Programa de prueba

```
cad HOLA = “hola mundo”
mostrar HOLA
```

Derivación vertical a derecha

```
programa
listaDeclaraciones listaSentencias
listaDeclaraciones sentIO
listaDeclaraciones sentSalida
listaDeclaraciones mostrar salida
listaDeclaraciones mostrar identificador
listaDeclaraciones mostrar HOLA
sentDeclaracion mostrar HOLA
tipoDato Identificador valorAsignado mostrar HOLA
tipoDato Identificador = ValorAAsignar mostrar HOLA
tipoDato Identificador = Cadena mostrar HOLA
tipoDato Identificador = “hola mundo” mostrar HOLA
tipoDato HOLA = “hola mundo” mostrar HOLA
cad HOLA = “hola mundo” mostrar HOLA
```

05/ CONCLUSIONES

Logramos desarrollar la sintaxis de un lenguaje de programación capaz de realizar operaciones de suma y multiplicación con números enteros, así como solicitar datos al usuario, mostrar datos y cadenas de texto por pantalla. Empleamos expresiones regulares para definir las diferentes categorías léxicas requeridas, como palabras reservadas, operadores, entre otros. Además, establecimos categorías sintácticas utilizando la notación BNF.